Anomaly Detection

전 재 욱

Embedded System 연구실
성균관대학교

Outline

- Problem motivation
- Gaussian distribution
- Algorithm
- Developing and evaluating an anomaly detection system
- Anomaly detection vs. supervised learning
- Choosing what features to use
- Multivariate Gaussian distribution
- Anomaly detection using the multivariate Gaussian distribution

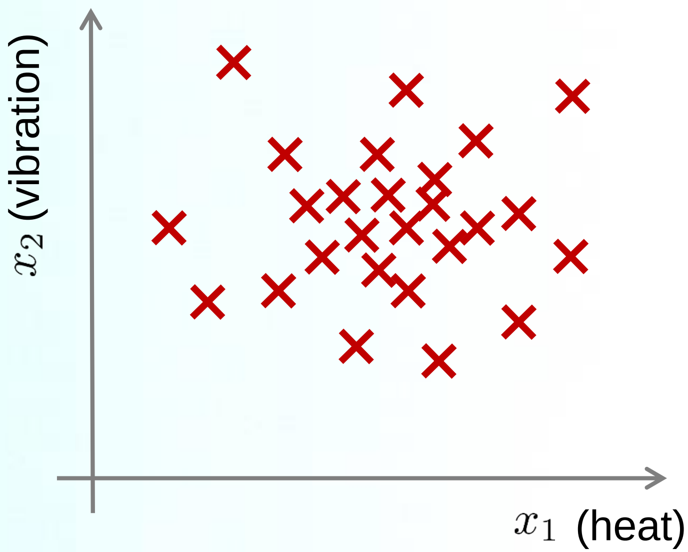
Outline

- Problem motivation
- Gaussian distribution
- Algorithm
- Developing and evaluating an anomaly detection system
- Anomaly detection vs. supervised learning
- Choosing what features to use
- Multivariate Gaussian distribution
- Anomaly detection using the multivariate Gaussian distribution

Anomaly Detection Example

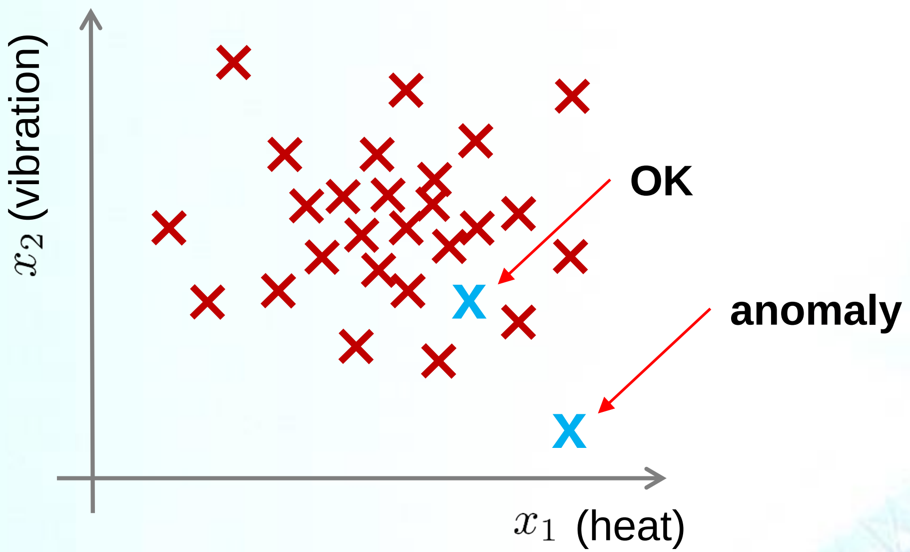
- Aircraft engine features
  - x_1: heat generated
  - x_2: vibration intensity

- Dataset: $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$



Anomaly Detection Example

- Given a new engine,
  - an anomaly detection method is used
    to see if the new engine is anomalous
    (when compared to the previous engines)



Anomaly Detection

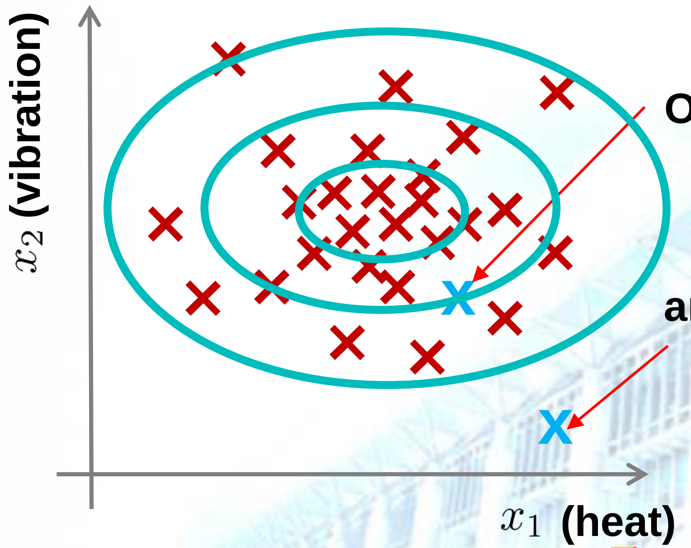
- More formally
  - We have a dataset which contains **normal** (data)
    - How can we ensure they are normal data?
      - It is up to us.
    - In reality, it is OK if this dataset contains a few which are NOT normal.
  - Using this dataset as a reference point,

    we can check whether other examples are normal or **anomalous**

Density Estimation

- Using our training dataset: $\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$
  - A model $p(x)$ can be built
    - $p(x)$: the probability that one example x is normal

- Having built a model, given x_{test}
  - If $p(x_{test}) < \varepsilon$ ➔ flag this as an anomaly
  - If $p(x_{test}) \geq \varepsilon$ ➔ this is OK
    - ε is some threshold probability value which we define, depending on how sure we want to be



Anomaly Detection Example

- Fraud detection:
  - $x^{(i)}$: features of user's activities
  - Model $p(x)$ from data

  - Identify unusual users by checking which have $p(x_{test}) < \varepsilon$

Anomaly Detection Example

- Fraud detection:
  - Users have activity associated with them, such as
    - Length on time on-line
    - Location of login
    - Spending frequency
  - Using this data,

    we can build a model $p(x)$ of what normal users' activity is like
  - What is the probability of "normal" behavior?
  - Identify unusual users by sending their data through the model

    (i.e. check $p(x_{test}) < \varepsilon$)
    - Flag up anything that looks a bit weird
    - Automatically block cards/transactions

Anomaly Detection Example

- Monitoring computers in data center
  - If many machines are in a cluster,
  - $x^{(i)}$: features of user i's activities
    - Computer features of machine
      - x_1 = memory use
      - x_2 = number of disk accesses/sec
      - x_3 = CPU load
  - In addition to the measurable features, our own complex features can also be defined
    - x_4 = CPU load/network traffic
  - If we see an anomalous machine (i.e. check $p(x_{test}) < \varepsilon$)
    - Maybe about to fail
    - Look at replacing bits from it

Outline

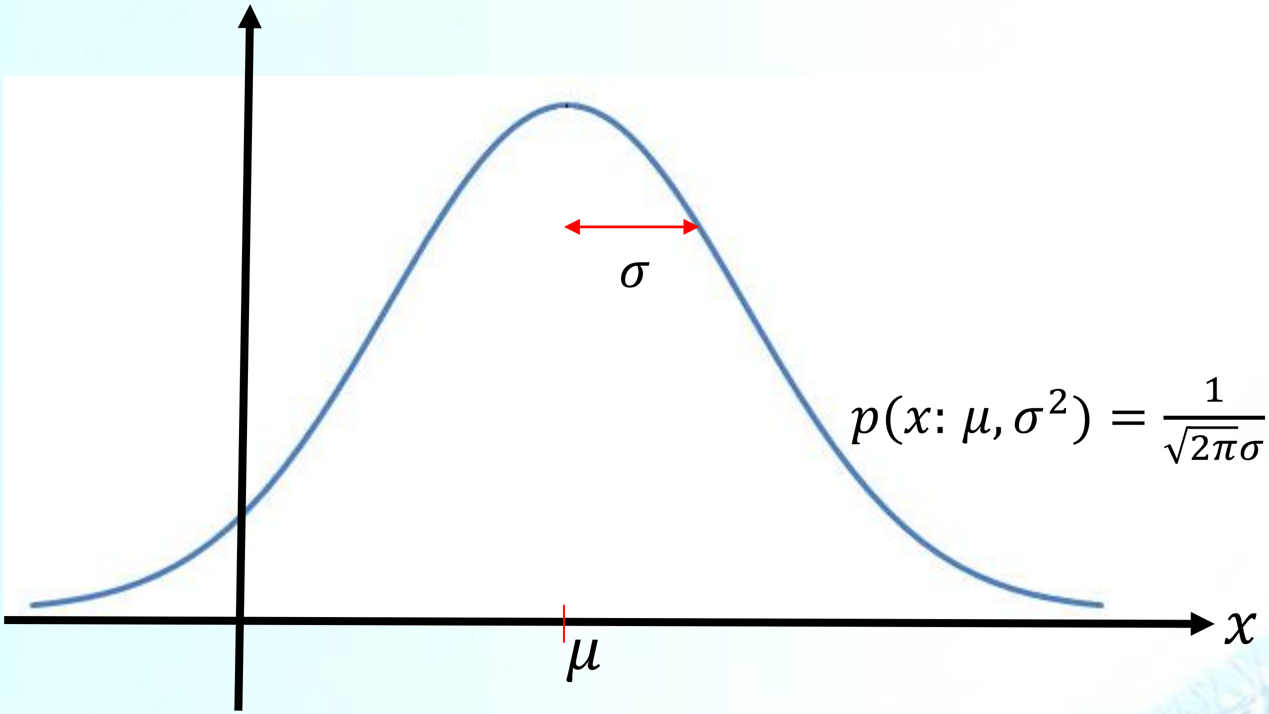
- Problem motivation
- Gaussian distribution
- Algorithm
- Developing and evaluating an anomaly detection system
- Anomaly detection vs. supervised learning
- Choosing what features to use
- Multivariate Gaussian distribution
- Anomaly detection using the multivariate Gaussian distribution

Gaussian (Normal) Distribution

- Say $x \in R$
  - If x is a distributed Gaussian with mean μ and variance σ^2

$$x \sim N(\mu, \sigma^2)$$

"distributed as"

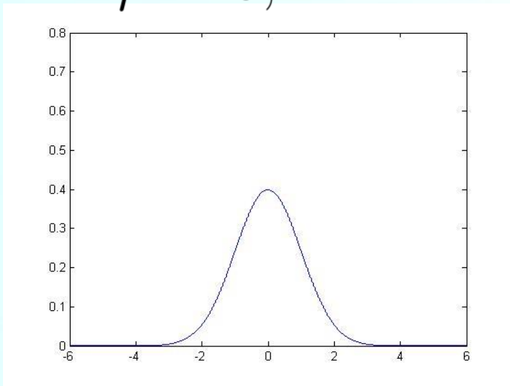


$$p(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

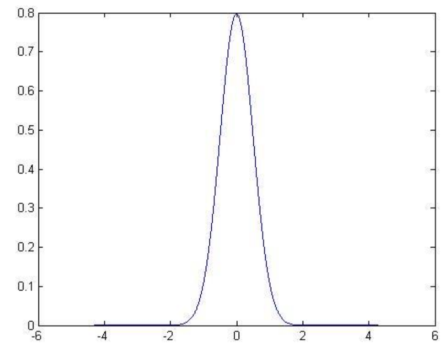
Gaussian Distribution Example

- Area under a Gaussian distribution is always 1
  - But width changes as standard deviation changes

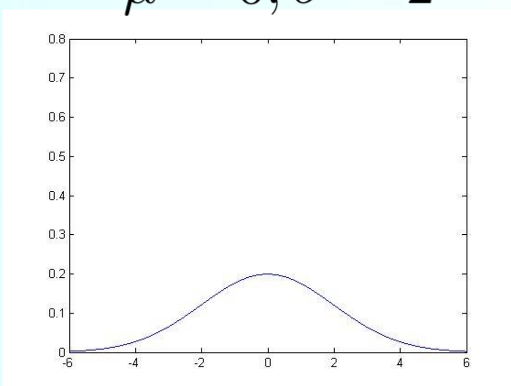
$$\mu = 0, \sigma = 1$$



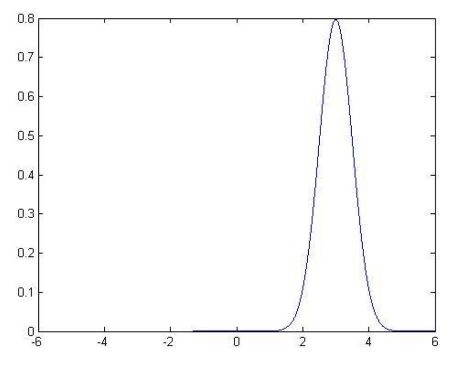
$$\mu = 0, \sigma = 0.5$$



$$\mu = 0, \sigma = 2$$



$$\mu = 3, \sigma = 0.5$$

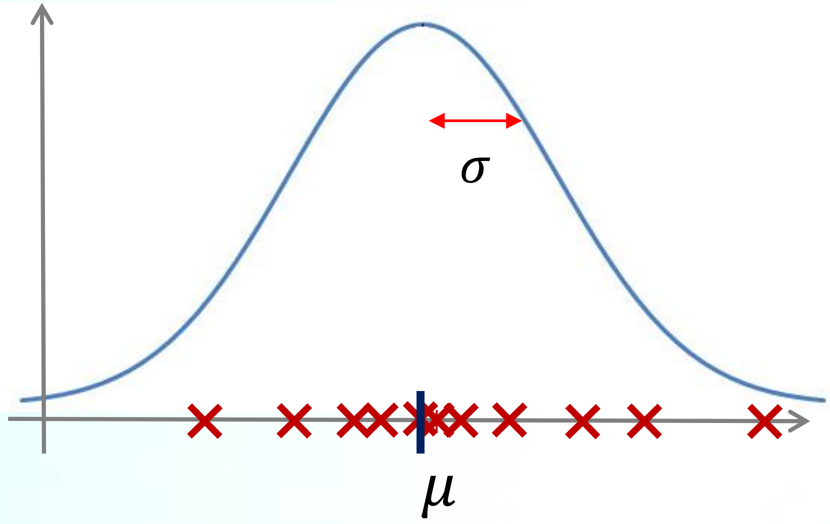


Parameter Estimation Problem

- Given a dataset: $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$, $\quad x^{(i)} \in R$
  - we can plot these data on the x axis
- Do these data come from a Gaussian?
  - Can we estimate the distribution for the given dataset?

Parameter Estimation Problem

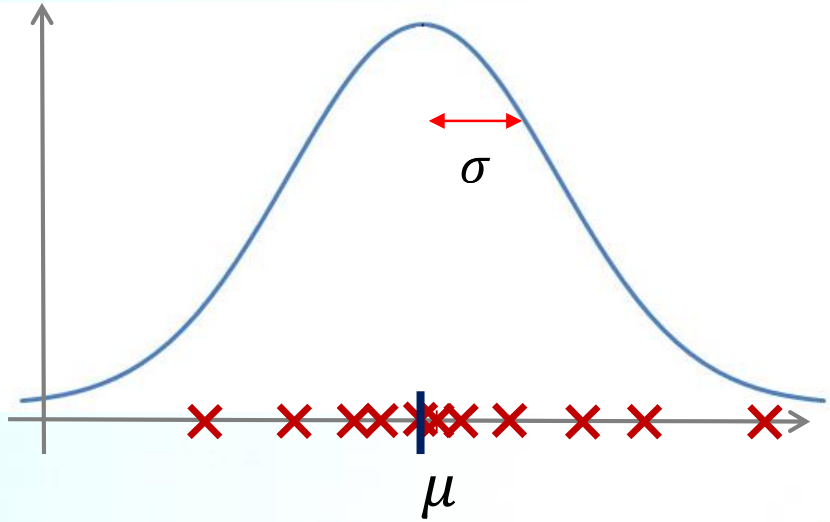
- A possible Gaussian could be the following curve
  - It seems like a reasonable fit
    - Data seems like a higher probability of being in the central region, lower probability of being further away



Parameter Estimation Problem

- Estimation of μ and σ^2
  - μ: average of data, $\mu = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}$
  - σ^2: variance of data, $\sigma^2 = \frac{1}{m}\sum_{i=1}^{m}\left(x^{(i)} - \mu\right)^2$

Maximum likelihood estimation for μ and σ^2



Outline

- Problem motivation
- Gaussian distribution
- Algorithm
- Developing and evaluating an anomaly detection system
- Anomaly detection vs. supervised learning
- Choosing what features to use
- Multivariate Gaussian distribution
- Anomaly detection using the multivariate Gaussian distribution

Density Estimation

- Unlabeled training set: $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$, $x^{(i)} \in R^n$
  - Each example is an n feature vector.

- Model $p(x)$ from the data set
  - What are high probability features and low probability features
  - x is a vector
    - So model $p(x)$ as

$$p(x) = p(x_1; \mu_1, \sigma_1^2) * p(x_2; \mu_2, \sigma_2^2) * \ \ldots * p(x_n; \mu_n, \sigma_n^2) = \prod_{j=1}^{n} p\left(x_j; \mu_j, \sigma_j^2\right)$$

(➔Independence assumption)

  - Here, we assume each feature is distributed according to a Gaussian distribution
    - $x_i \sim N\left(x_1; \mu_i, \sigma_i^2\right)$
    - $p\left(x_j; \mu_j, \sigma_j^2\right)$
      - The probability of feature x_j (given μ_j and σ_j^2) using a Gaussian distribution

Density Estimation

- The previous equation

$$p(x) = p(x_1; \mu_1, \sigma_1^2) * p(x_2; \mu_2, \sigma_2^2) * \ \dots * p(x_n; \mu_n, \sigma_n^2) = \prod_{j=1}^{n} p\left(x_j; \mu_j, \sigma_j^2\right)$$

makes an **independence assumption** for the features

- although algorithm works if features are independent or not
  - If features are tightly linked,
    we should be able to do some dimensionality reduction anyway.

Anomaly Detection Algorithm

- Choose features x_i that we think might be indicative of anomalous examples.

- Given a training set $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$
  - Fit parameters $\mu_1, \ldots, \mu_n$, $\sigma_1, \ldots, \sigma_n$

$$\mu_j = \frac{1}{m}\sum_{i=1}^{m} x_j^{(i)} , \qquad \sigma_j = \frac{1}{m}\sum_{i=1}^{m}\left(x_j^{(i)} - \mu_j\right)^2$$

- Given new example x, compute $p(x)$:

$$p(x) = \prod_{j=1}^{n} p\left(x_j : \mu_j, \sigma_j^2\right) = \prod_{j=1}^{n} \frac{1}{\sqrt{2\pi}\sigma_j} \exp\left(-\frac{\left(x_j - \mu_j\right)^2}{2\sigma_j^2}\right)$$

- Anomaly if $p(x) < \varepsilon$

Anomaly Detection Algorithm

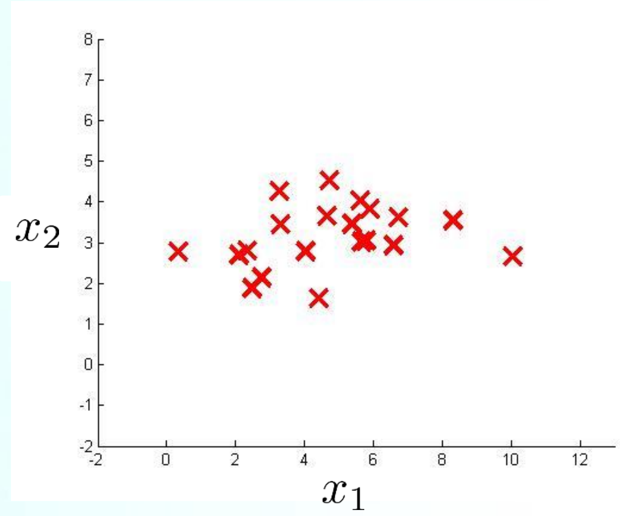
- Chose features
  - Try to come up with features which might help identify something anomalous - may be unusually large or small values
    - More generally, choose features which describe the general properties
    - This is nothing unique to anomaly detection
      - It is just the idea of building a sensible feature vector

- Fit parameters for a given training set
  - Determine parameters for each example: μ_i and σ_i^2
    - Variance and mean for each feature are calculated

- Compute $p(x)$
  - If $p(x)$ is very small,
    it has very low chance for x to be "normal"

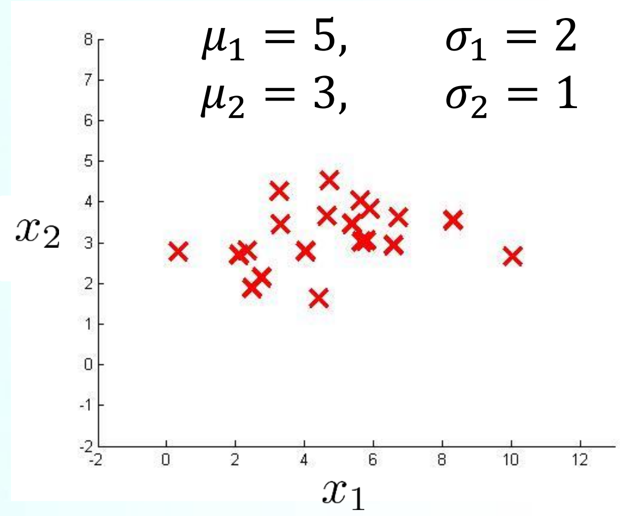
Anomaly Detection Example

- x_1
- x_2



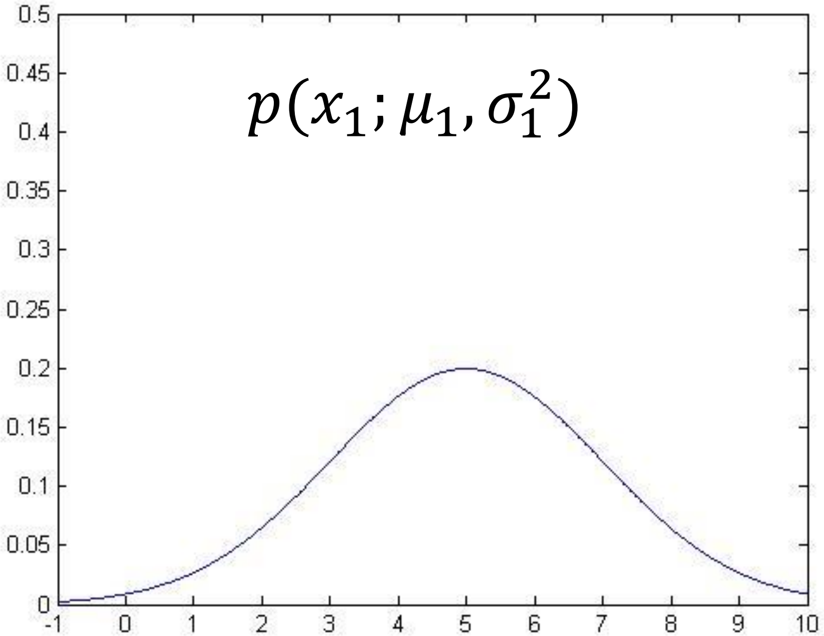
Anomaly Detection Example

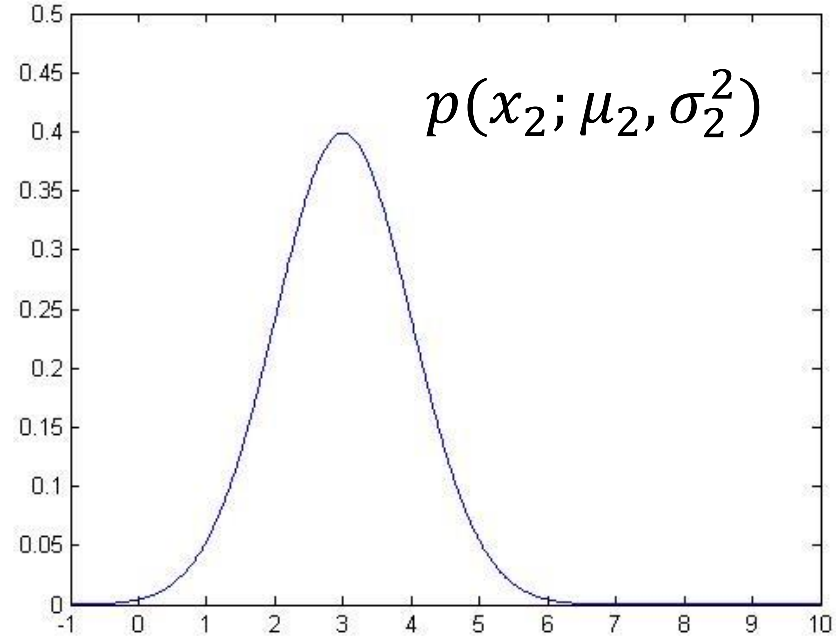
- x_1
  - Mean: 5 , Standard deviation: 2
- x_2
  - Mean: 3 , Standard deviation: 1



Anomaly Detection Example

- Gaussian for x_1 and x_2

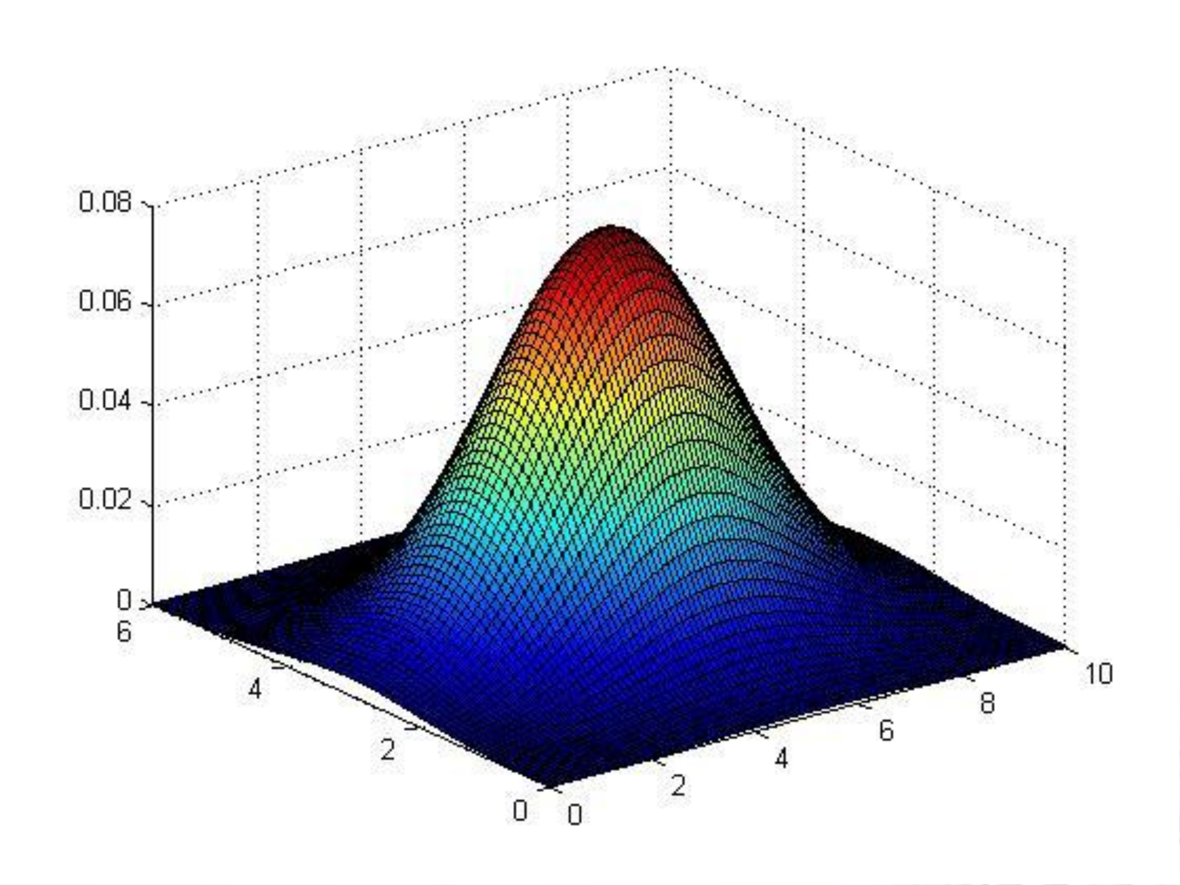




Anomaly Detection Example

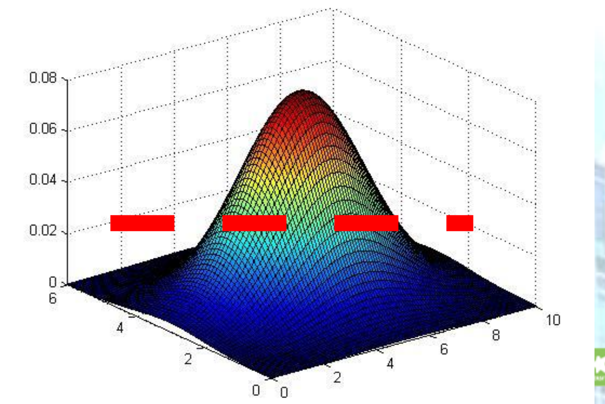
- Plot for $p(x_1;\mu_1,\sigma_1^2) \times p(x_2;\mu_2,\sigma_2^2)$
  - The height of the surface is the probability

$$p(x) = p(x_1;\mu_1,\sigma_1^2) * p(x_2;\mu_2,\sigma_2^2)$$



Anomaly Detection Example

- Check if a test value is anomalous
  - For example, set $\varepsilon = 0.02$
  - Given two new data $x_{test}^{(1)}$ and $x_{test}^{(2)}$,
    - If $p\left(x_{test}^{(1)}\right) = 0.0426$ ➔ normal ($0.0426 \geq \varepsilon$)
    - If $p\left(x_{test}^{(2)}\right) = 0.0021$ ➔ anomalous ($0.0021 < \varepsilon$)
  - Considering the probability $p(x)$ as the surface height,
    - All values above a certain height are normal,
    - all the values below that threshold are probably anomalous



Outline

- Problem motivation
- Gaussian distribution
- Algorithm
- Developing and evaluating an anomaly detection system
- Anomaly detection vs. supervised learning
- Choosing what features to use
- Multivariate Gaussian distribution
- Anomaly detection using the multivariate Gaussian distribution

Importance of Real-Number Evaluation

- When developing a learning algorithm (choosing features, etc.),
  - making decisions is much easier

    if we have a way of evaluating our learning algorithm.
    - which gives us a single number

- Easier to evaluate our algorithm

  if a **single number** is given to show

if changes we made improved or worsened an algorithm's performance

(Depending on the inclusion of one extra feature or not)

Importance of Real-Number Evaluation

- Assume we have some labeled data, of anomalous and non-anomalous examples.
  - ($y = 0$ if normal, $y = 1$ if anomalous).

- Training set is the collection of normal examples
  - OK even if we have a few anomalous data examples
  - $x^{(1)},\ x^{(2)}, \dots ,\ x^{(m)}$

- Define cross validation set and test set
  - $\left(x_{cv}^{(1)}, y_{cv}^{(1)}\right), \left(x_{cv}^{(2)}, y_{cv}^{(2)}\right), \dots, \left(x_{cv}^{(m_{cv})}, y_{cv}^{(m_{cv})}\right)$
  - $\left(x_{test}^{(1)}, y_{test}^{(1)}\right), \left(x_{test}^{(2)}, y_{test}^{(2)}\right), \dots, \left(x_{test}^{(m_{test})}, y_{test}^{(m_{test})}\right)$
    - For both cross validation and test sets,
      - assume we can include a few examples which have anomalous examples

Aircraft Engines Motivating Example

- Engines
  - Have 10,000 good (normal) engines
    - OK even if a few bad ones are here
    - Lots of $y = 0$
  - 20 flawed engines (anomalous)
    - Typically when $y = 1$ have 20~50

- Split into
  - Training set: 6,000 good engines ($y = 0$)
  - CV set: 2,000 good engines($y = 0$), 10 anomalous($y = 1$)
  - Test set: 2000 good engines($y = 0$), 10 anomalous($y = 1$)
  - Ratio is 3:1:1

Aircraft Engines Motivating Example

- Engines
  - Have 10,000 good (normal) engines
    - OK even if a few bad ones are here
    - Lots of $y = 0$
  - 20 flawed engines (anomalous)
    - Typically when $y = 1$ have 20~50

- Alternative

  Exactly same

  - Training set: 6,000 good engines ($y = 0$)
  - CV set: 4,000 good engines($y = 0$), 10 anomalous($y = 1$)
  - Test set: 4,000 good engines($y = 0$), 10 anomalous($y = 1$)
  - ➔ NOT good practice
    - Should use different data in CV and test set.

Algorithm Evaluation

- Fit model $p(x)$ on training set $\{x^{(1)}, \dots, x^{(m)}\}$

- On a cross validation/test example x , predict

$$y = \begin{cases} 1 & if \quad p(x) < \varepsilon \quad (anomaly) \\ 0 & if \quad p(x) \geq \varepsilon \quad (normal) \end{cases}$$

- Possible evaluation metrics:
  - True positive, false positive, false negative, true negative
  - Precision/Recall
  - F_1-score
  - (classification would be NOT good because $y = 0$ is very common)

- Can also use cross validation set to choose parameter ε

Algorithm Evaluation

- Can also use cross validation set to choose parameter ε
  - If we have CV set, we can see how varying ε effects various evaluation metrics
    - Then pick the value of ε which maximizes the score on our CV set

Outline

- Problem motivation
- Gaussian distribution
- Algorithm
- Developing and evaluating an anomaly detection system
- Anomaly detection vs. supervised learning
- Choosing what features to use
- Multivariate Gaussian distribution
- Anomaly detection using the multivariate Gaussian distribution

Anomaly Detection vs. Supervised Learning

- Anomaly Detection
  - Very small number of positive examples ($y = 1$).
    - (0-20 is common)
  - Large number of negative ($y = 0$) examples.
  - Many different "types" of anomalies.
    - Hard for any algorithm to learn from positive examples what the anomalies look like;
    - future anomalies may look nothing like any of the anomalous examples we have seen so far.

- Supervised learning
  - Large number of positive and negative examples
  - Enough positive examples for algorithm to get a sense of what positive examples are like,
    - future positive examples likely to be similar to ones in training set.

Anomaly Detection

- Very small number of positive examples
  - Save positive examples just for CV and test set
  - Consider using an anomaly detection algorithm
  - Not enough data to "learn" positive examples
- Have a very large number of negative examples
  - Use these negative examples for $p(x)$ fitting
  - Only need negative examples for this

Anomaly Detection

- Many different "types" of anomalies
  - *<u>Hard for an algorithm to learn from positive examples</u>* when anomalies may look nothing like one another
    - So anomaly detection does not know what they look like, but knows what they *do not* look like
  - When we looked at SPAM email,
    - Many types of SPAM
    - For the spam problem, usually enough positive examples
      - So this is why we usually think of SPAM as supervised learning

Anomaly Detection vs. Supervised Learning

- Anomaly detection
  - Fraud detection
  - Manufacturing (e.g. aircraft engines)
  - Monitoring machines in a data center
  - …

- Supervised learning
  - Email spam classification
  - Weather prediction (sunny/rainy/etc)
  - Cancer classification
  - …

Anomaly Detection

- Application
  - Fraud detection
    - Many ways of fraud
    - If we are a major on line retailer/very subject to attacks,
      we sometimes might shift to supervised learning
  - Manufacturing (e.g. aircraft engines)
    - If we make HUGE volumes,
      we may have enough positive data ➔ make supervised
      - Means we make an assumption about the kinds of errors we are going to see
  - Monitoring machines in a data center
  - …

Supervised Learning

- Reasonably large number of positive and negative examples

- Have enough positive examples to give your algorithm the opportunity to see what they look like

- Application
  - Email/SPAM classification
  - Weather prediction
  - Cancer classification

Outline

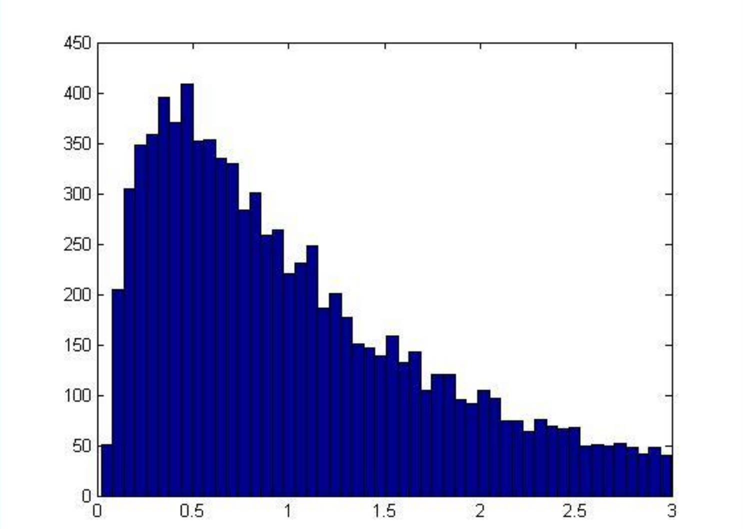
- Problem motivation
- Gaussian distribution
- Algorithm
- Developing and evaluating an anomaly detection system
- Anomaly detection vs. supervised learning
- Choosing what features to use
- Multivariate Gaussian distribution
- Anomaly detection using the multivariate Gaussian distribution

Choosing Features to Use

- Huge effect on an anomaly detection
  - which features are used

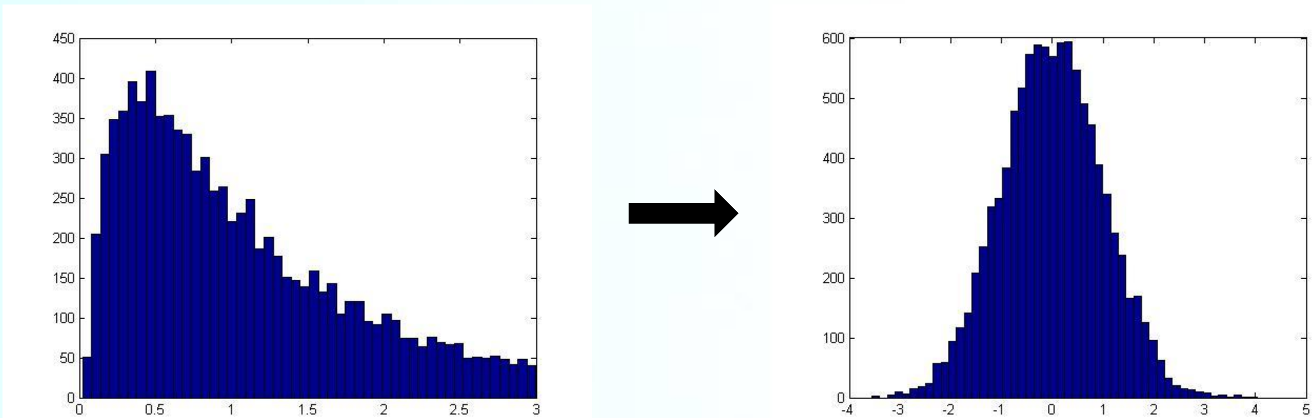
Choosing Features to Use

- Non-Gaussian features
  - Plot a histogram of data
    to check it has a Gaussian description
    - Often still works if data is non-Gaussian
    - Use **hist** command to plot histogram
  - Non-Gaussian data might look like this



Choosing Features to Use

- Transforming non-Gaussian data into a Gaussian data
  - Different transformation of the data to make it look more Gaussian
    - A log transformation of the data
      - For some feature x_1, replace it with $\log(x_1)$



      - Or do $\log(x_1 + c)$
        - Add c to make it look as Gaussian as possible
      - Or do $x^{1/2}$
      - Or do $x^{1/3}$

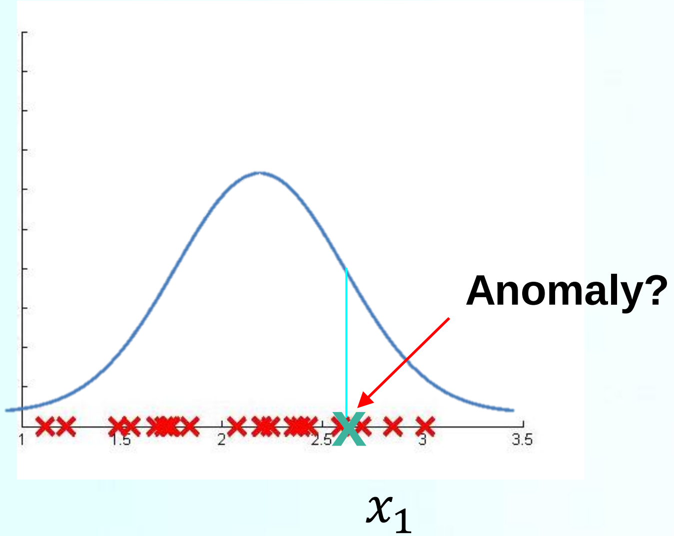
Error Analysis for Anomaly Detection

- Like supervised learning, error analysis procedure
  - Run algorithm on CV set
  - See which one it got wrong
  - Develop new features based on trying to understand *why* the algorithm got those examples wrong

Error Analysis for Anomaly Detection

- Want $p(x)$ large for normal examples x .
  $p(x)$ small for anomalous examples x .

- Most common problem:
  - $p(x)$ is comparable (say, both large)
    for normal and anomalous examples



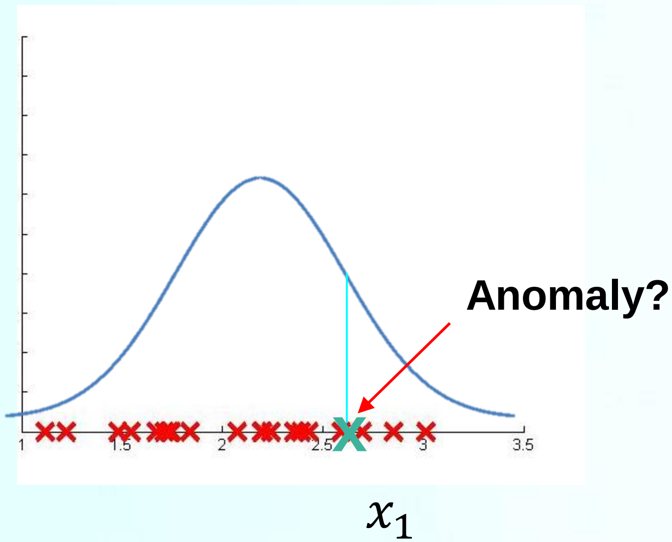
- Our anomalous value is sort of buried in it
  - Look at data - see what went wrong

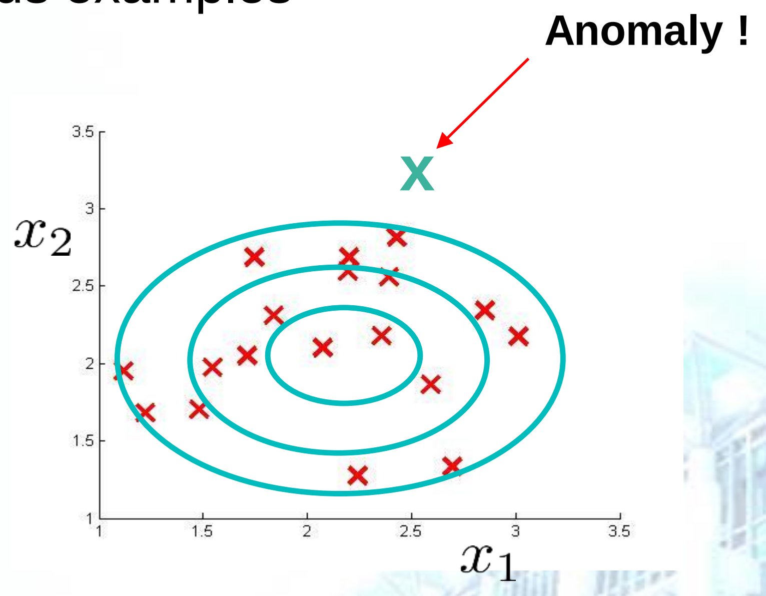
- Develop a new feature x_2 which can help distinguish further anomalous

Error Analysis for Anomaly Detection

- Want $p(x)$ large for normal examples x .
  $p(x)$ small for anomalous examples x .

- Most common problem:
  - $p(x)$ is comparable (say, both large) for normal and anomalous examples





Monitoring Computers in A Data Center

- Choose features that might take on unusually large or small values in the event of an anomaly.

  - x_1: memory use of computer
  - x_2: number of disk accesses/sec
  - x_3: CPU load
  - x_4: network traffic
  - …
  - x_5: (CPU load)/(network traffic)
  - x_6: $(\text{CPU load})^2/(\text{network traffic})$

Outline

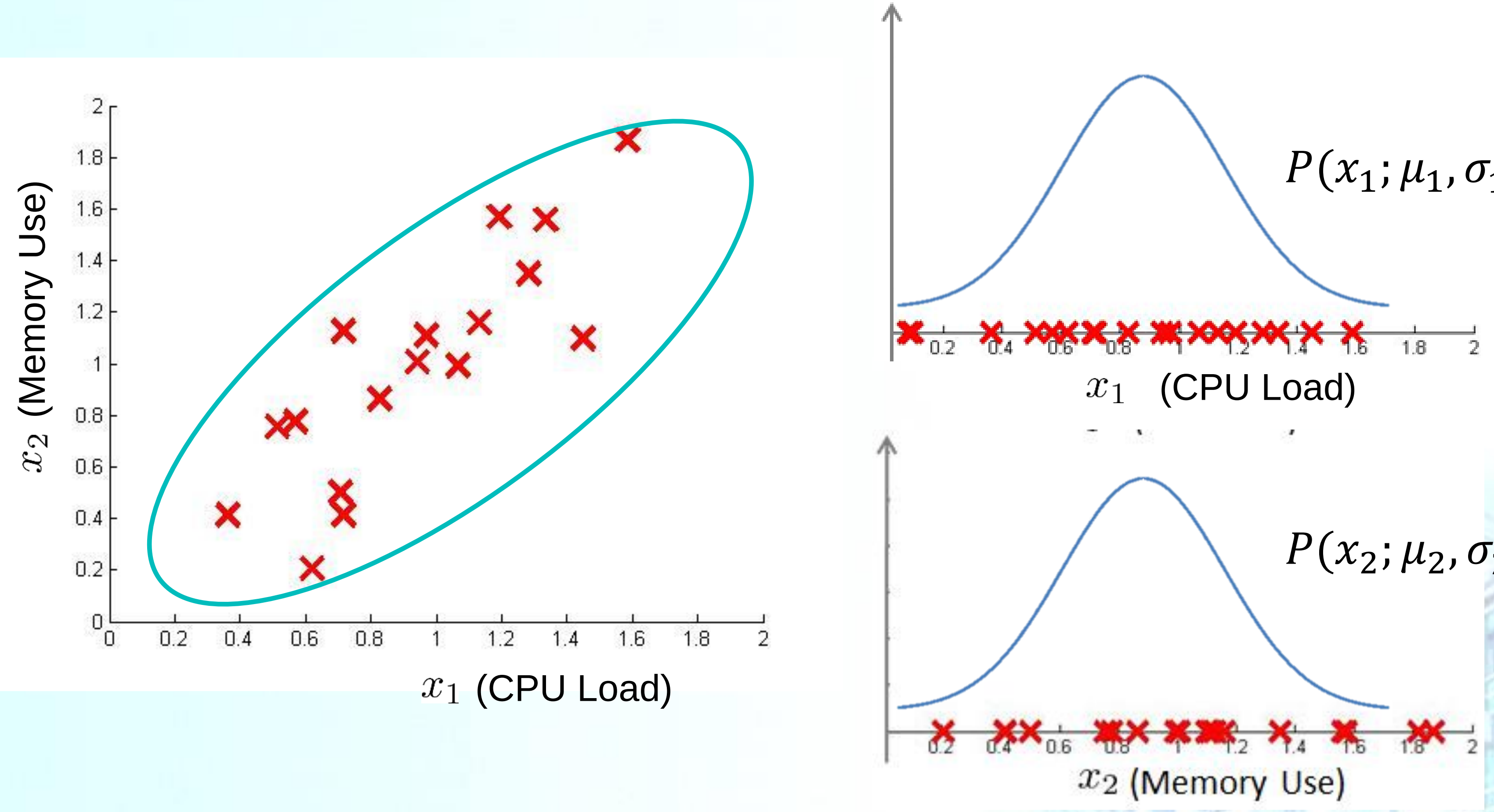
- Problem motivation
- Gaussian distribution
- Algorithm
- Developing and evaluating an anomaly detection system
- Anomaly detection vs. supervised learning
- Choosing what features to use
- Multivariate Gaussian distribution
- Anomaly detection using the multivariate Gaussian distribution

Multivariate Gaussian Distribution

- Multivariate Gaussian Distribution
  - SA slightly different technique which can sometimes catch some anomalies
    - which non-multivariate Gaussian distribution anomaly detection fails to

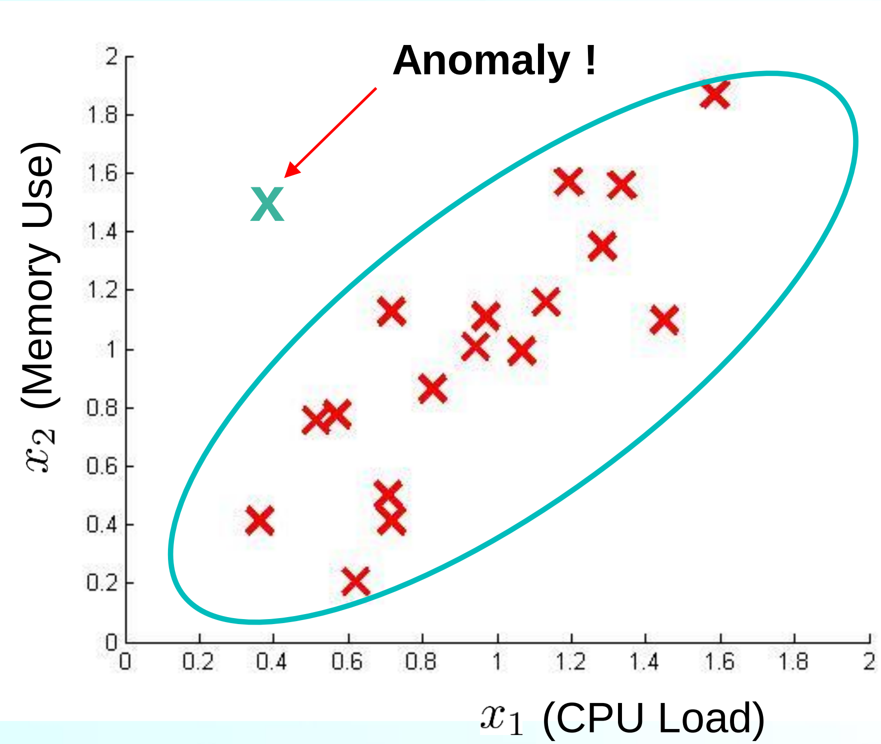
Multivariate Gaussian Distribution

- Unlabeled data looks like this
  - A Gaussian distribution to CPU load and memory use



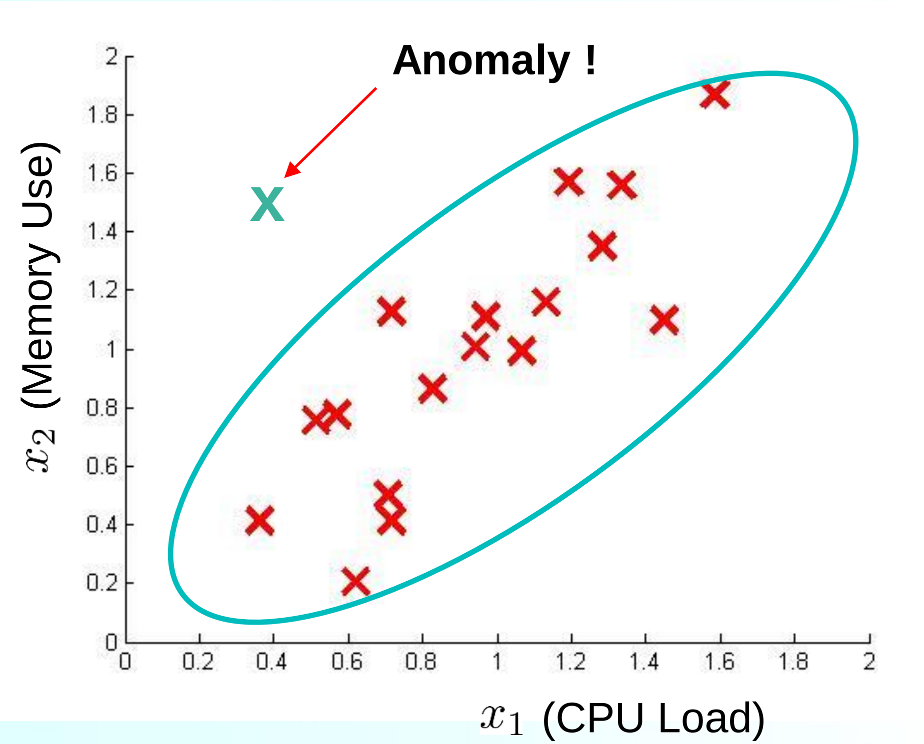
Multivariate Gaussian Distribution

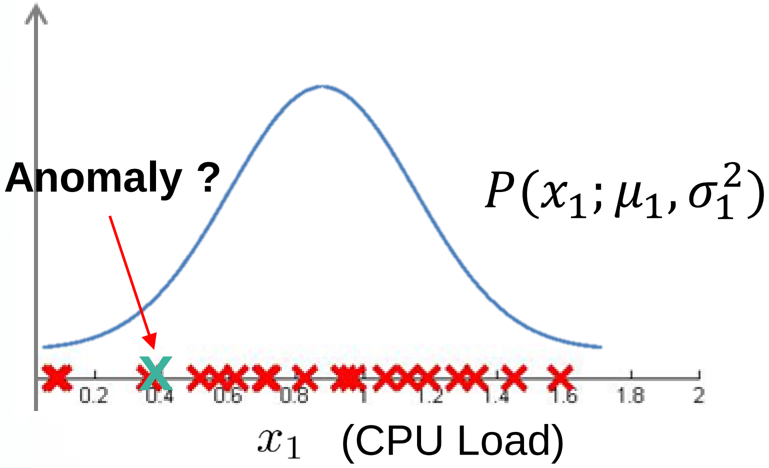
- One example in the test set
  - which looks like an anomaly (e.g. $x_1 = 0.4$, $x_2 = 1.5$)
    - memory use is high and CPU load is low

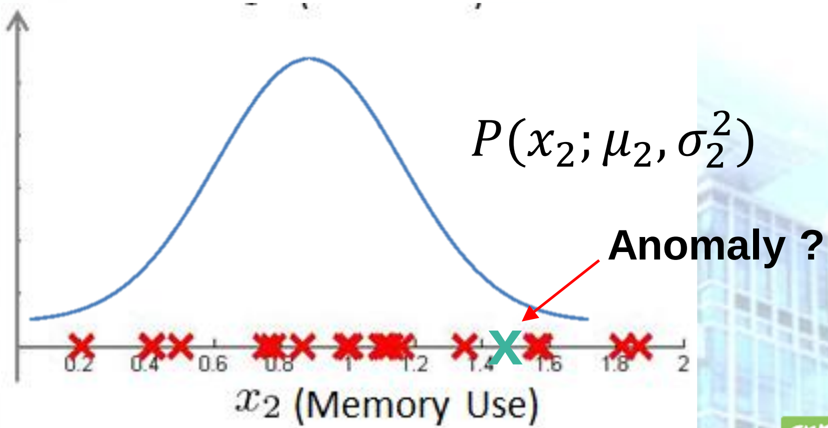


Multivariate Gaussian Distribution

- Problem
  - If we look at each feature individually, they are both acceptable

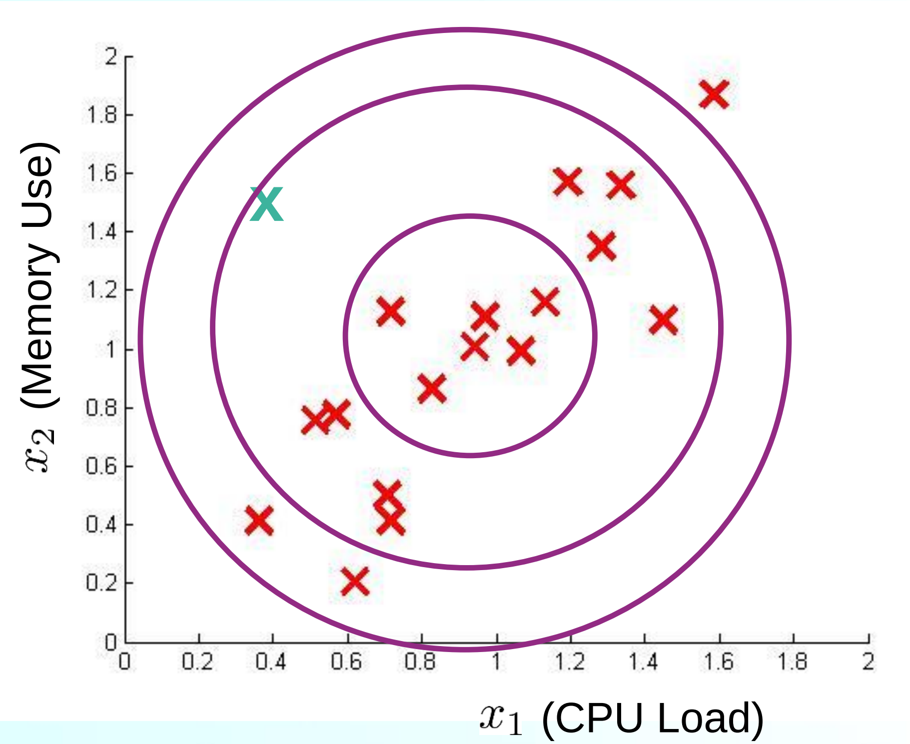


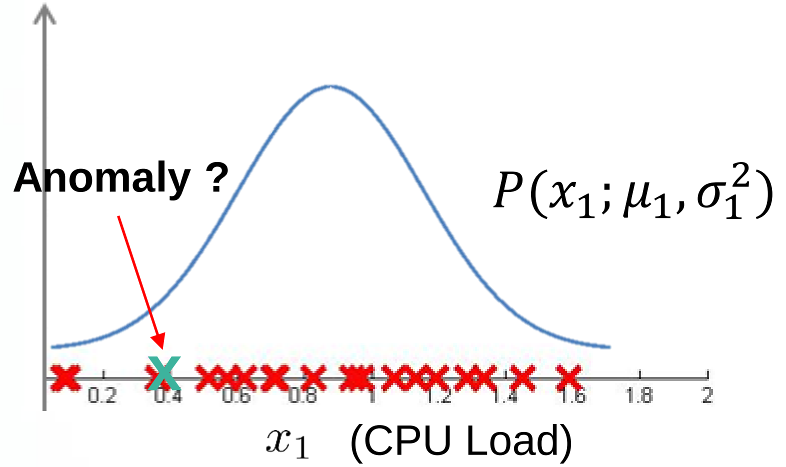


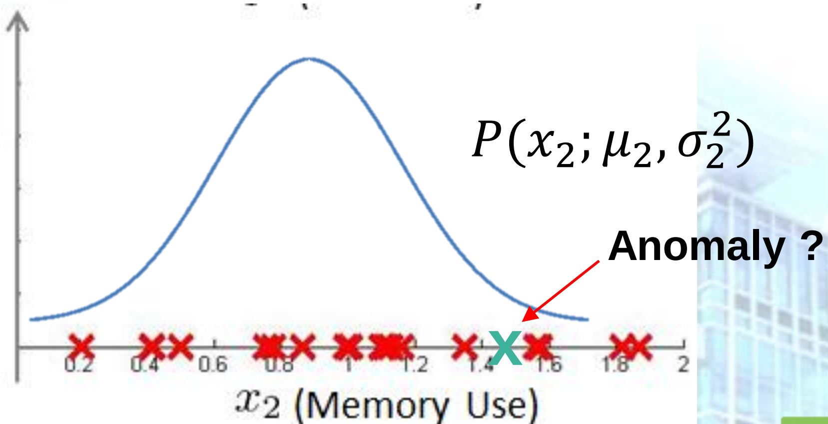


Multivariate Gaussian Distribution

- Problem
  - This is because our function makes probability prediction in concentric circles around the means of both

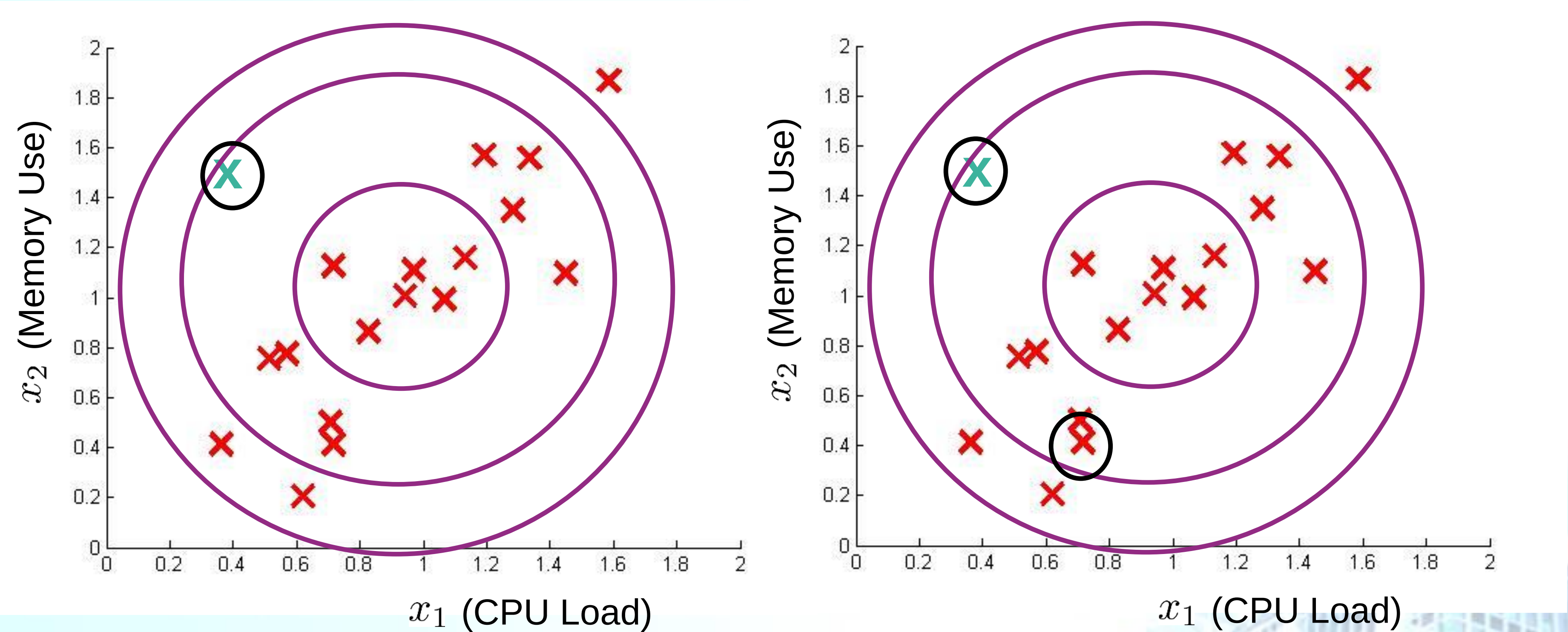






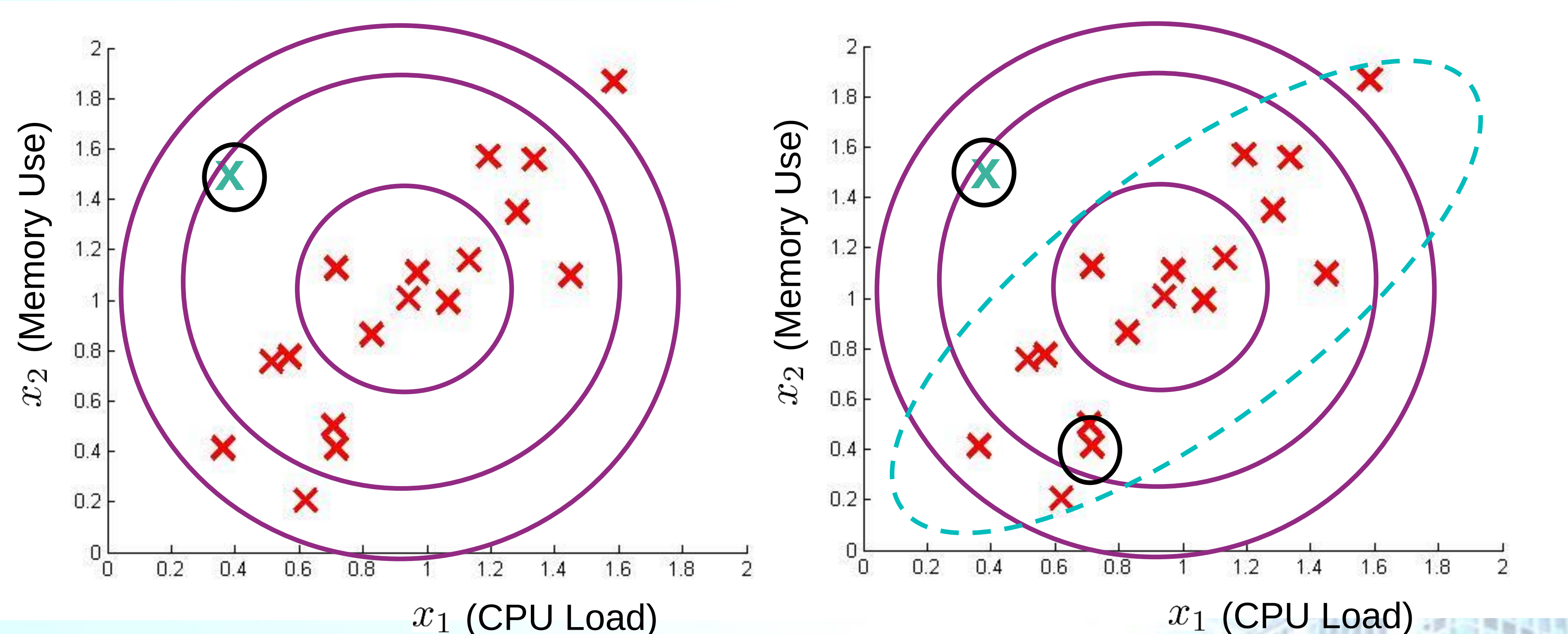
Multivariate Gaussian Distribution

- Problem
  - Probability of the two black circled examples is basically the same, even though we can clearly see the green one as an outlier



Multivariate Gaussian Distribution

- Problem
  - Probability of the two black circled examples is basically the same, even though we can clearly see the green one as an outlier



➔ To get around this, we develop the multivariate Gaussian distribution

Multivariate Gaussian Distribution

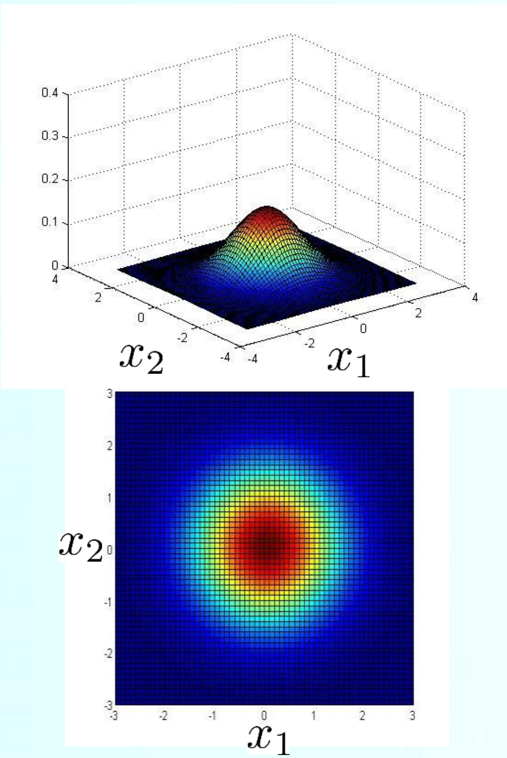
- Given $x \in R^n$,
  - Do not model $p(x_1), p(x_2), \dots, p(x_n)$ seperately.
  - Model $p(x)$ in one go.
    - Parameters: $\mu \in R^n$, $\Sigma \in R^{n \times n}$ (covariance matrix)

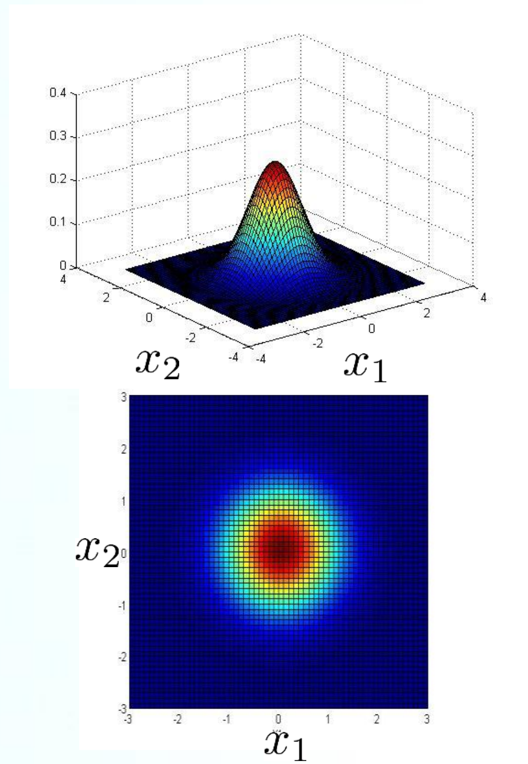
- Multivariate Gaussian Distribution

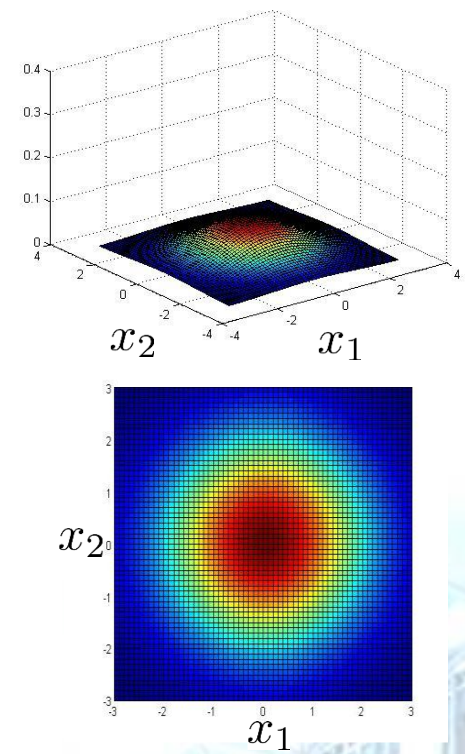
$$p(x; \mu, \Sigma) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right)$$

Multivariate Gaussian Distribution

$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Sigma = \begin{bmatrix} 0.6 & 0 \\ 0 & 0.6 \end{bmatrix} \qquad \mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Sigma = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

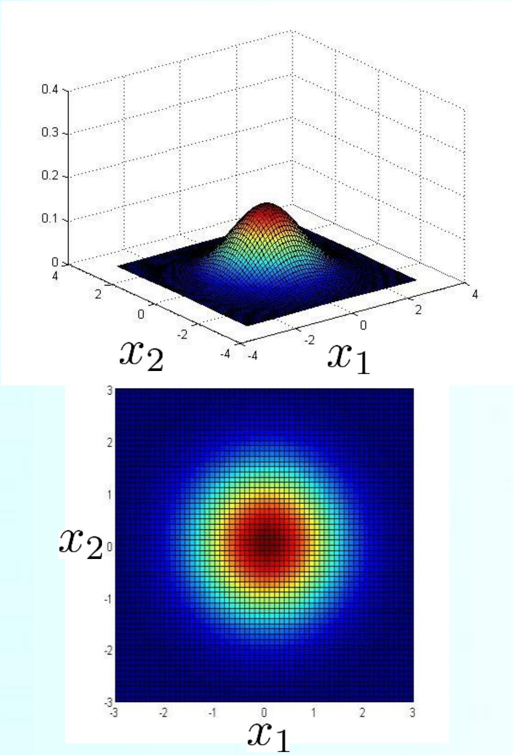


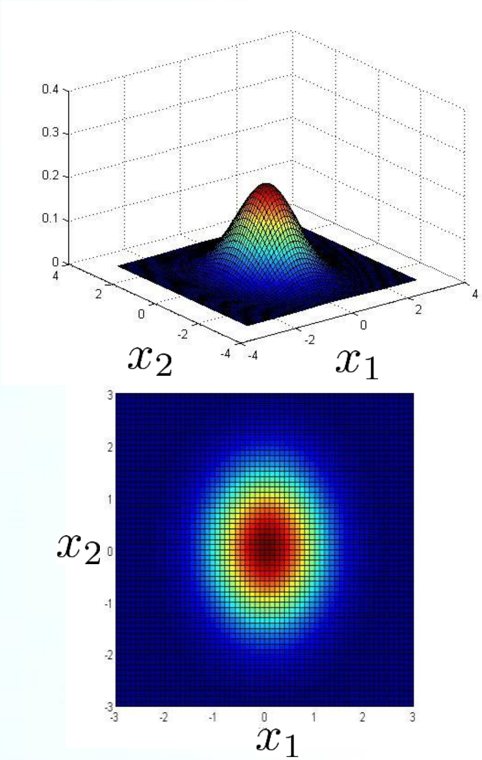


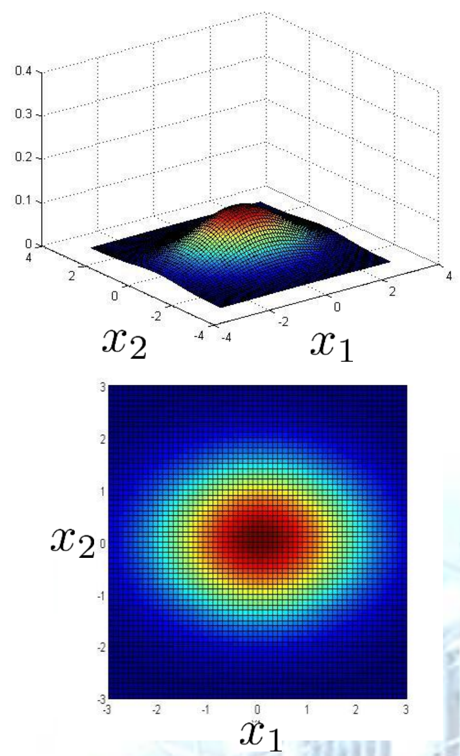


Multivariate Gaussian Distribution

$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ $\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Sigma = \begin{bmatrix} 0.6 & 0 \\ 0 & 1 \end{bmatrix}$ $\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Sigma = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$

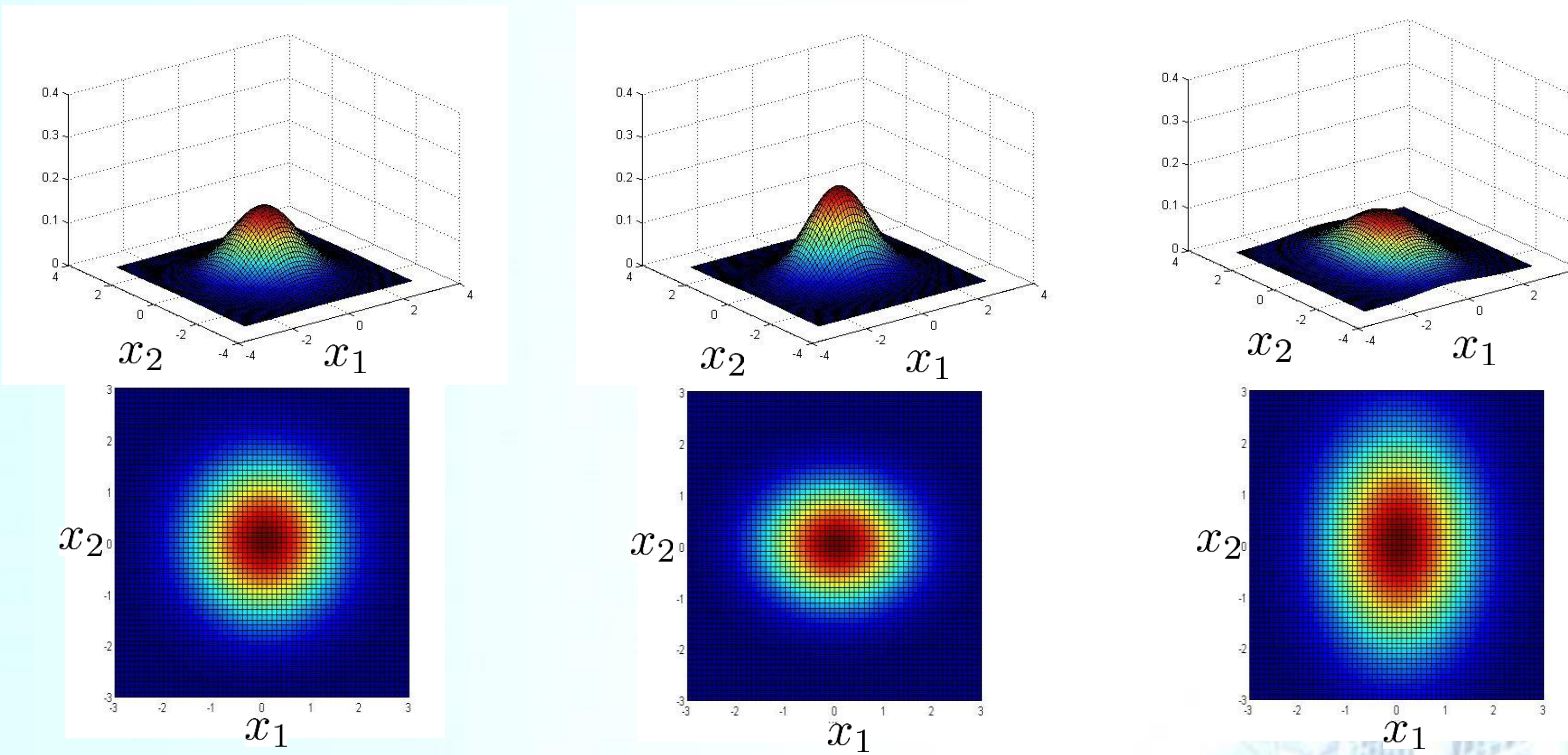






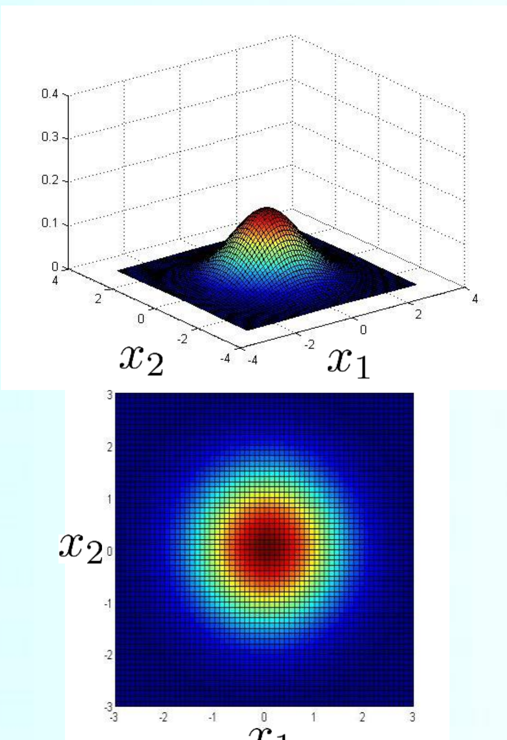
Multivariate Gaussian Distribution

$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 0.6 \end{bmatrix} \qquad \mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$

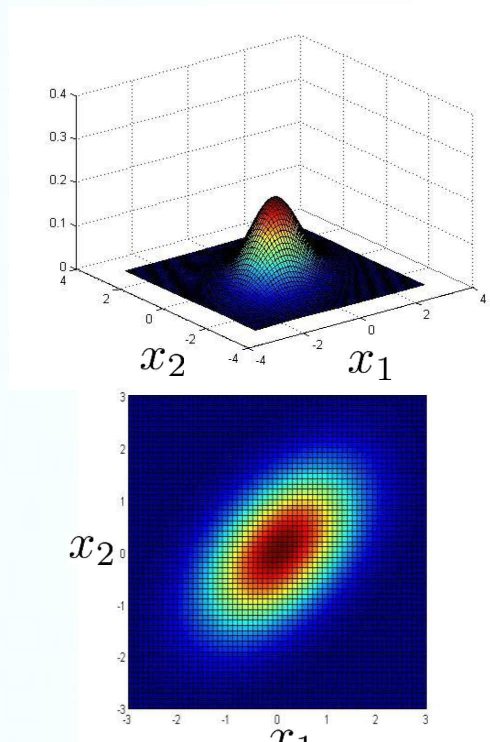


Multivariate Gaussian Distribution

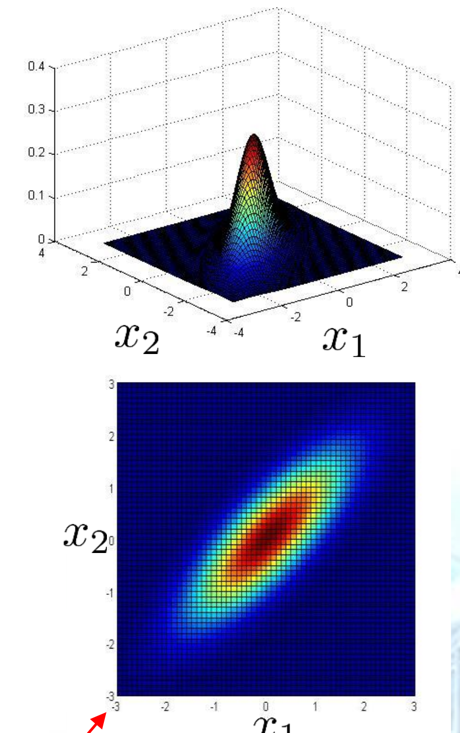
$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$



$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$



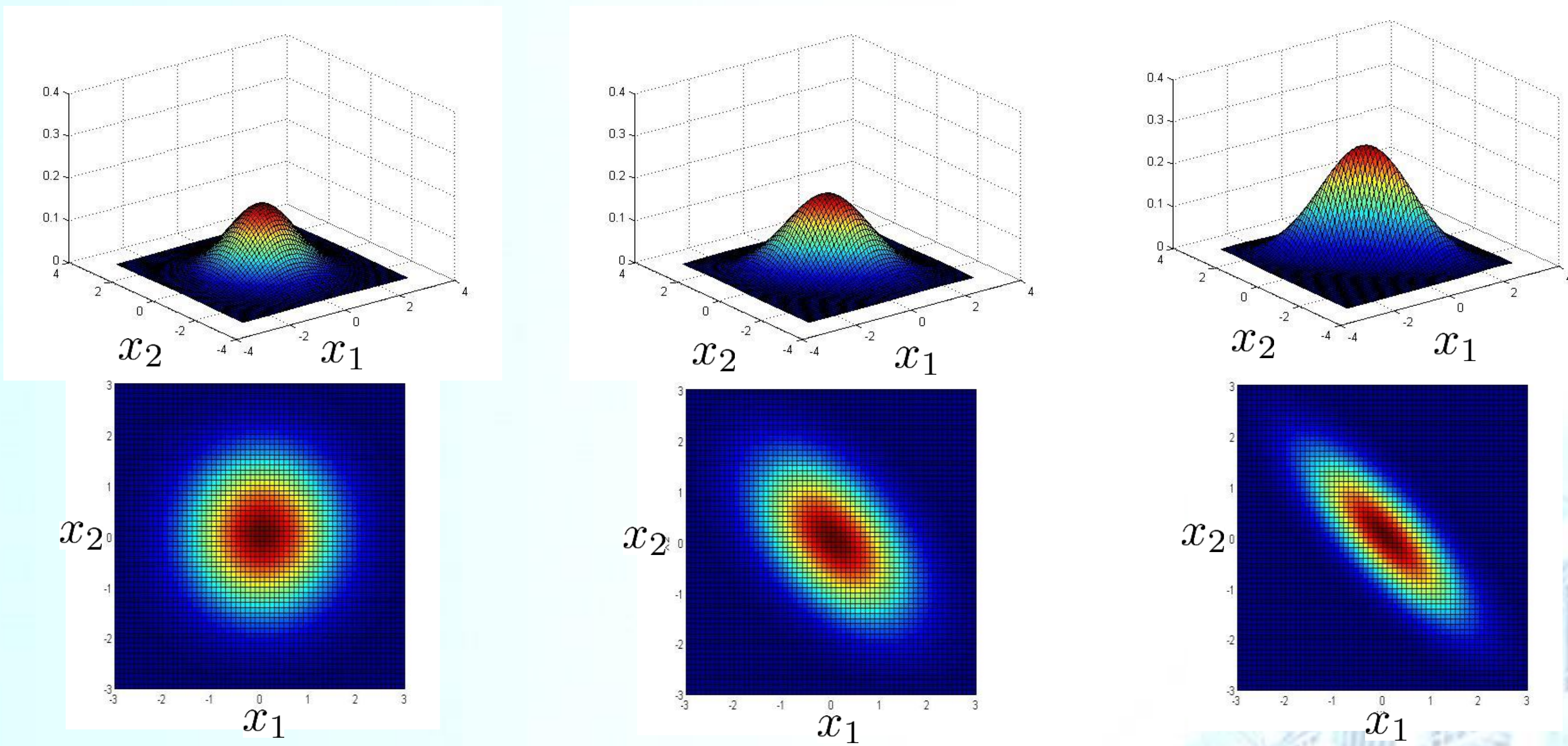
$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Sigma = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$



this example gives a very tall thin distribution, shows a strong positive correlation

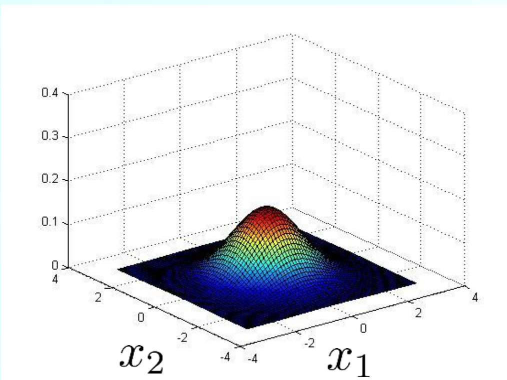
Multivariate Gaussian Distribution

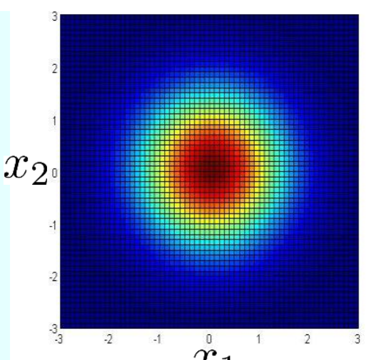
$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Sigma = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix} \qquad \mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Sigma = \begin{bmatrix} 1 & -0.8 \\ -0.8 & 1 \end{bmatrix}$$

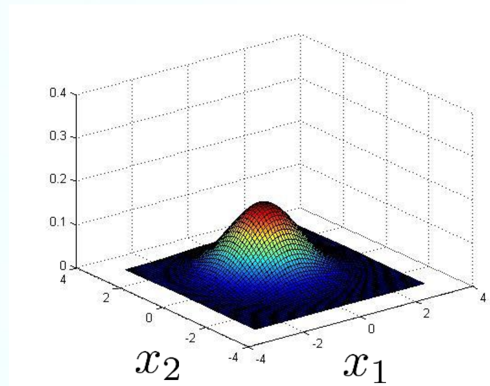


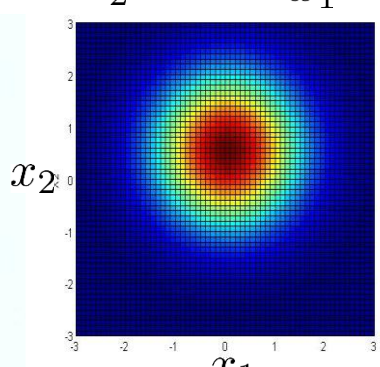
Multivariate Gaussian Distribution

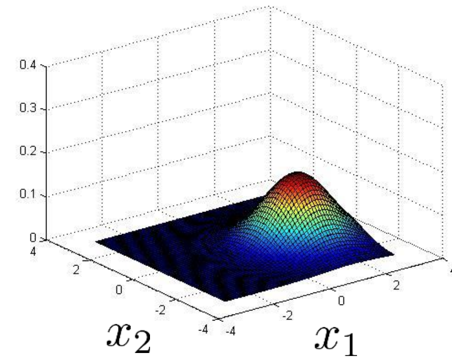
$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ $\mu = \begin{bmatrix} 0 \\ 0.5 \end{bmatrix}, \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ $\mu = \begin{bmatrix} 1.5 \\ -0.5 \end{bmatrix}, \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

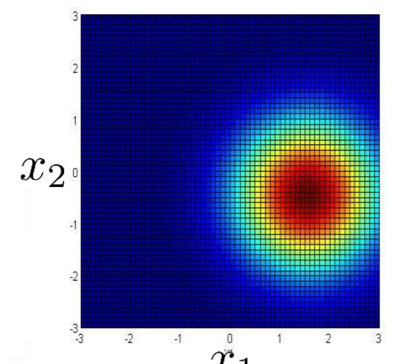












Outline

- Problem motivation
- Gaussian distribution
- Algorithm
- Developing and evaluating an anomaly detection system
- Anomaly detection vs. supervised learning
- Choosing what features to use
- Multivariate Gaussian distribution
- Anomaly detection using the multivariate Gaussian distribution

Multivariate Gaussian Distribution

- Multivariate Gaussian Distribution

  Parameters: $\mu \in R^n$, $\Sigma \in R^{n \times n}$ (covariance matrix)

$$p(x; \mu, \Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right)$$

- Parameter fitting:
  - Given training set $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$,

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}, \qquad \Sigma = \frac{1}{m}\sum_{i=1}^{m}\left(x^{(i)} - \mu\right)\left(x^{(i)} - \mu\right)^T$$

Anomaly Detection with The Multivariate Gaussian

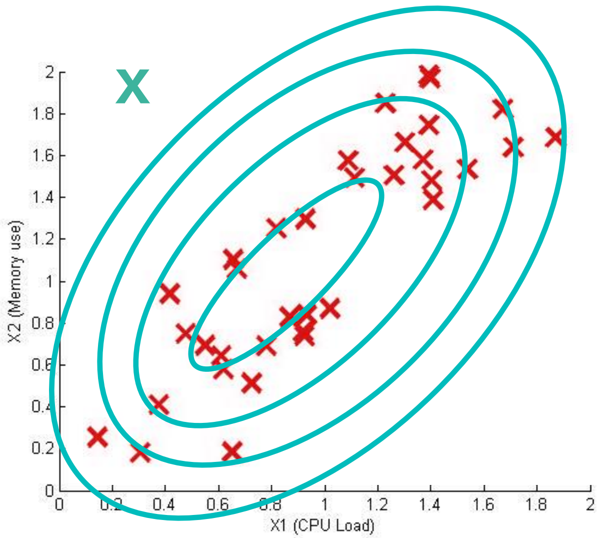
- Fit model $p(x)$ by setting

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}, \ \Sigma = \frac{1}{m}\sum_{i=1}^{m}(x^{(i)} - \mu)(x^{(i)} - \mu)^T$$

- Given a new example x_{test}, compute

$$p(x_{test}; \mu, \Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x_{test} - \mu)^T \Sigma^{-1}(x_{test} - \mu)\right)$$

Flag an anomaly if $p(x_{test}) < \varepsilon$

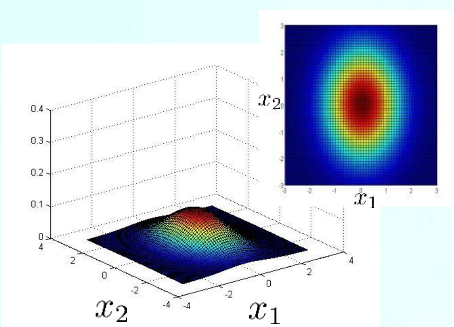


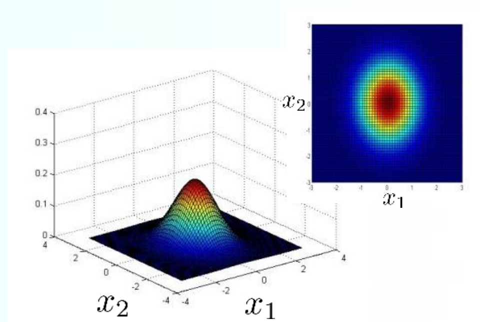
Relationship to Original Model

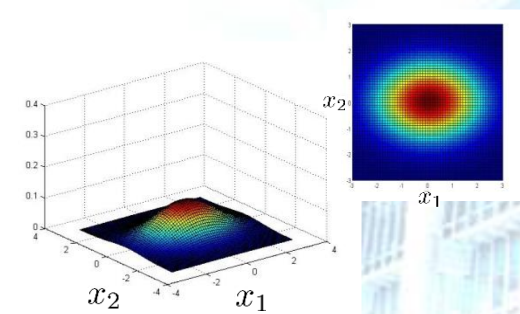
- Original model
  - $p(x) = p(x_1, \mu_1, \sigma_1^2) * p(x_2, \mu_2, \sigma_2^2) * \cdots * p(x_n, \mu_n, \sigma_n^2)$
    - corresponds to multivariate Gaussian where the Gaussians' contours are axis aligned
      - Has this constraint

        that the covariance matrix Σ as ZEROs on the non-diagonal values
    - $p(x; \mu, \Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right)$

where $\Sigma = \begin{bmatrix} \sigma_1^2 & & & \mathbf{0} \\ & \sigma_2^2 & & \\ & & \cdots & \\ \mathbf{0} & & & \sigma_n^2 \end{bmatrix}$







Original Model vs. Multivariate Gaussian

- Original Model

$$p(x) = p(x_1, \mu_1, \sigma_1^2) * p(x_2, \mu_2, \sigma_2^2) * \cdots * p(x_n, \mu_n, \sigma_n^2)$$

Manually create features to capture anomalies where x_1, x_2 take unusual combinations of values. (e.g. $x_3 = \frac{x_1}{x_2}$)

- Computationally cheaper (alternatively, scales better to large n)

- OK even if m (training set size) is small

- Multivariate Gaussian

$$p(x; \mu, \Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$

- Automatically captures correlations btw features

- Computationally more expensive

- Must have $m > n$, or else Σ is non-invertible.

Original Model vs. Multivariate Gaussian

- Original Gaussian
  - Probably used more often
  - Manually create features to capture anomalies where x_1 and x_2 take unusual combinations of values
    - So need to make extra features
      - For example, $x_3 = \frac{x_1}{x_2} = \frac{CPU\ load}{memory}$
    - Much cheaper computationally
  - Scales much better to very large feature vectors
    - Even if $n = 100{,}000$, the original model works fine
  - Works well even with a small training set
    - For example, $m = 50,\ 100$
  - Because of the above factors,

    it is used more often

    because it really represents a optimized

    but axis-symmetric specialization of the general model

Original Model vs. Multivariate Gaussian

- Multivariate Gaussian
  - Used less frequently
  - Can capture feature correlation ➔So no need to create extra values
  - Less computationally efficient
    - Must compute inverse of $[n \times n]$ matrix
    - So lots of features are bad - makes this calculation very expensive
    - So if $n = 100{,}000$, multivariate Gaussian is not very good
  - Needs for $m > n$
    - i.e. (number of examples) > (number of features)
    - If this is not true, then we have a singular matrix (non-invertible)
      - So should be used only in $m \gg n$
  - If you find the matrix is non-invertible,
    - $m < n$
      - So use original simple model
    - Redundant features (i.e. linearly dependent)
      - i.e. two features that are the same
      - If this is the case, use PCA or sanity check your data

References

- https://www.coursera.org/learn/machine-learning
- http://www.holehouse.org/mlclass/15_Anomaly_Detection.html